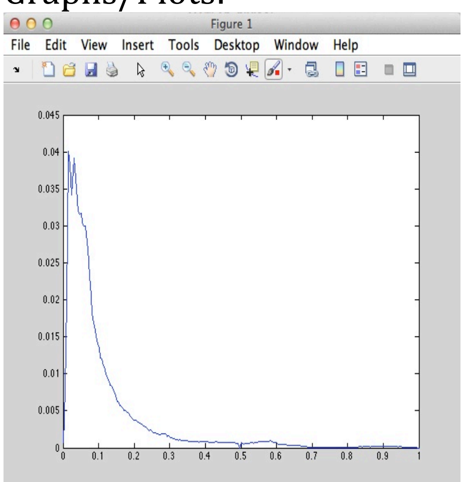
CS6640-Project 1

Atul Rungta
U0734304
atul.rungta@utah.edu

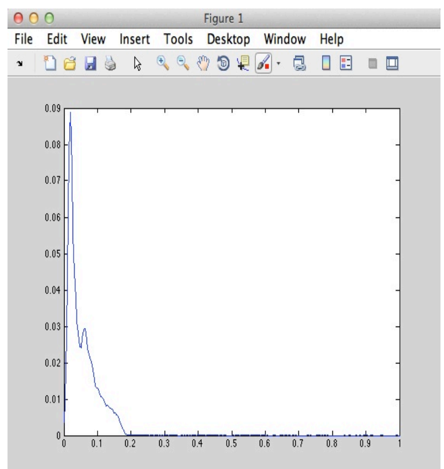
1. Histogram

The histogram was calculated by counting the number of pixels of each intensity value [0..255]. The relative frequency was found by dividing it by the total number of pixels (Calculated using noOfPixels = size(Image, 1) * size(Image, 2)).

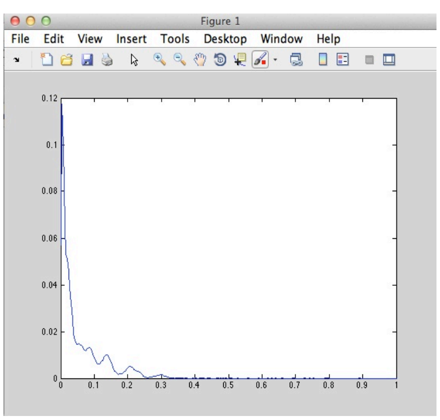
Graphs/Plots:



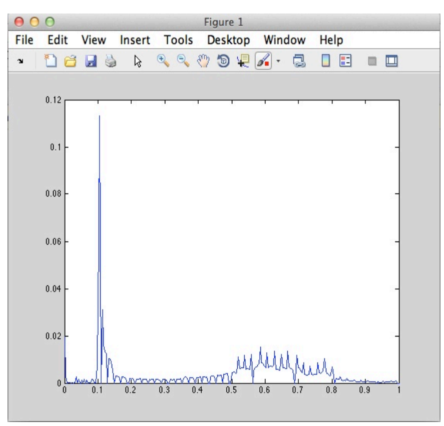
Crowd



University



Chang



Portal

The graphs were plot using MATLAB's plot function.

2. Histogram Equalization

The steps I followed for Histogram Equalization were taken from http://en.wikipedia.org/wiki/Histogram_equalization:

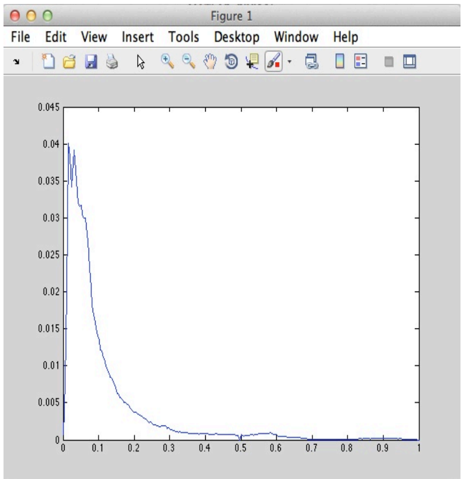
a. The count for each pixel value was calculated and stored.
b. The CDF was calculated.
c. The equalized histogram was calculated using the following formula:

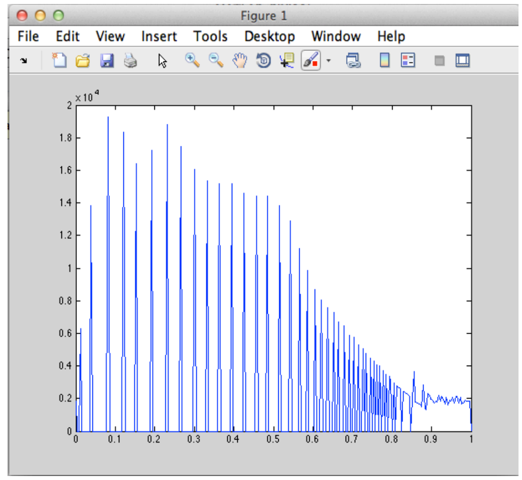
$$h(v) = round(((cdf(v) - cdf_{min}))/((M \times N) - cdf_{min})) \times (L - 1))$$

where
h(v) = the equalized histogram
cdf(v) = the value associated with the CDF(V)
cdf_{min} = the minimum CDF
M = width of the image
N = height of the image
L = total number of color levels.

Enhanced Image (Crowd):

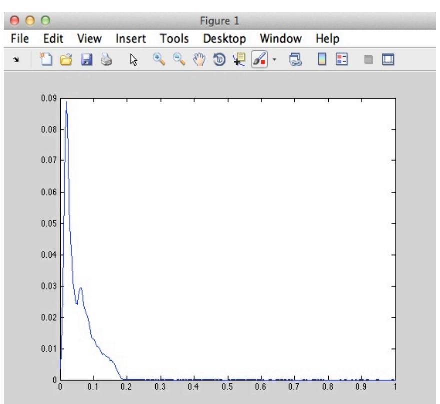
Histogram (Normal/Equalized):

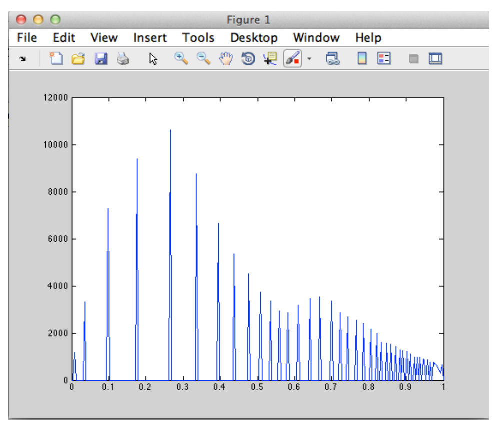




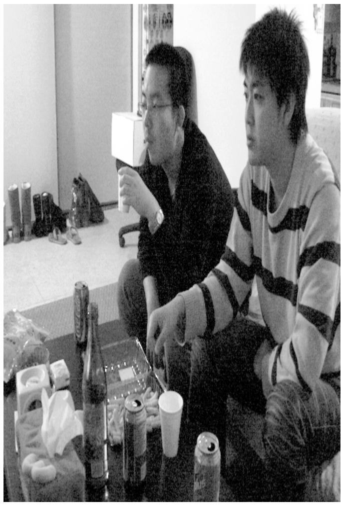
Enhanced Image (University):

Histogram (Normal/Equalized):

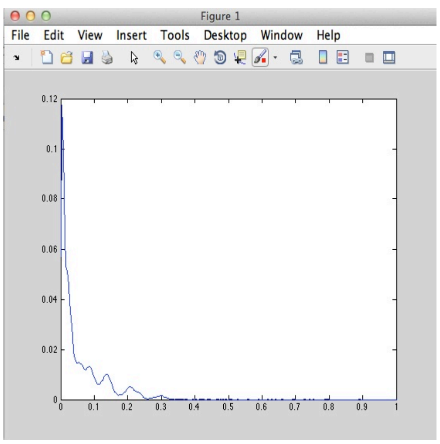


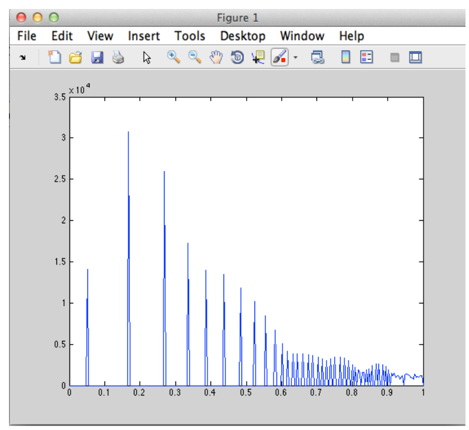


Enhanced Image (Chang):

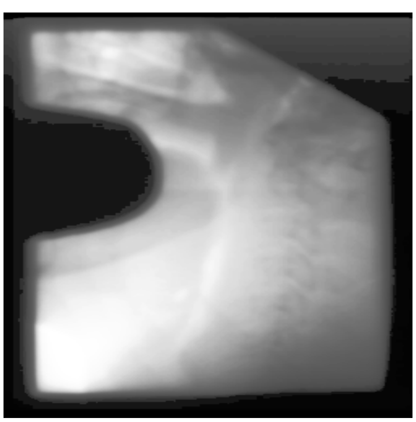


Histogram (Normal/Equalized):

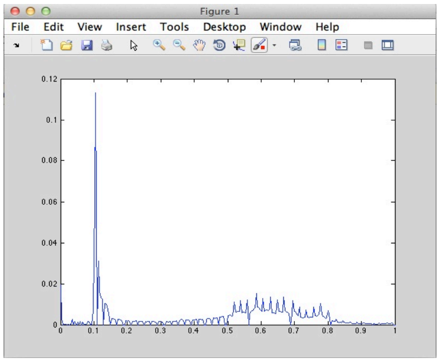


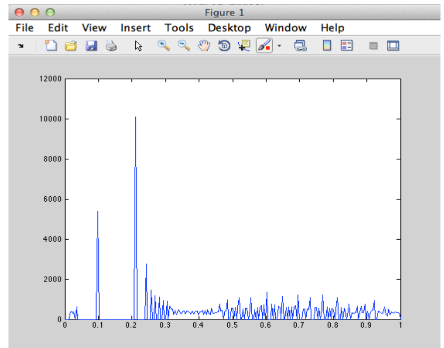


Enhanced Image (Portal):



Histogram (Normal/Equalized):





Other Images:

Original:

Enhanced:

Original:

Enhanced

Histogram Equalization is not effective for all images such as **Portal** because the distribution is already fairly well spread across all the intensity values. Histogram equalization seeks to enhance contrast by spreading the localized intensity values as evenly as possible, but since in this case, it's already present it does not do a good job of enhancing contrast.

For the blending function, I just used the original image and the equalized image and linearly blended them as specified:

- Get Equalized Image **EQI = histoeq(OriginalImage, n, min, max)**
- Take OriginalImage and blend according to **alpha * EQI + (1-alpha) * OriginalImage.**
- Save new image.

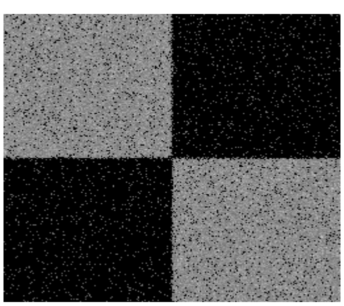
This is the result with **alpha = 0.5** applied to the Lena image:

Original

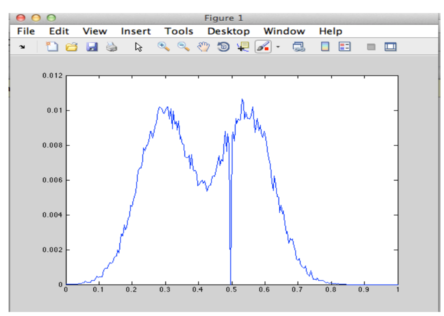
Enhanced (alpha=0.5)

As can be seen from the image above, with an alpha = 0.5 it is half way between fully equalized and the original one. In the fully equalized image shown the brightness is high causing saturation, which is reduced to some extent in the image seen above.

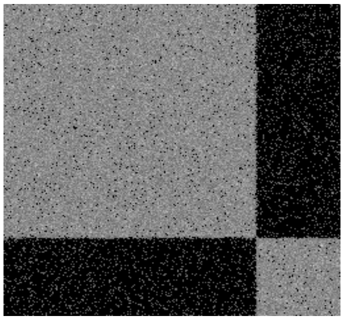
3. Segmentation by Thresholding



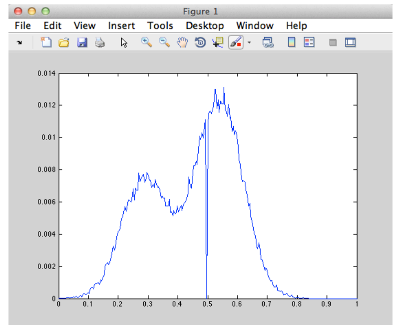
Checker1



For checker1 the threshold was found to be around 110, as can be seen in the histogram.

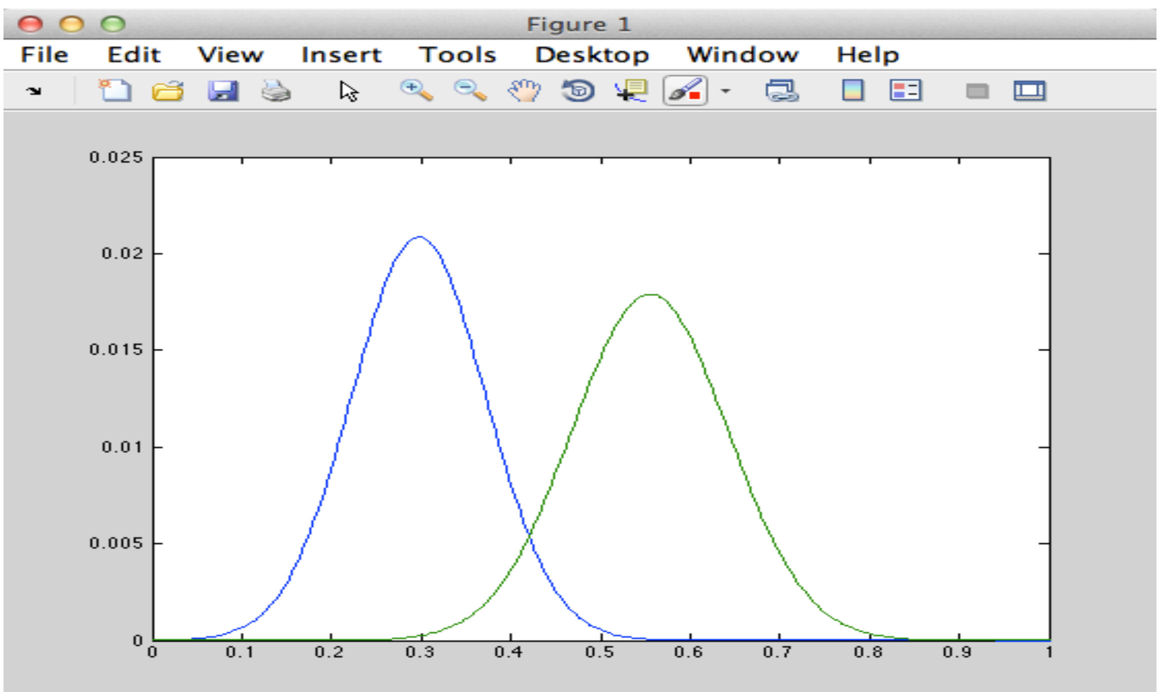


Checker2



For checker2 the threshold was found to be 98, as can be seen in the histogram.

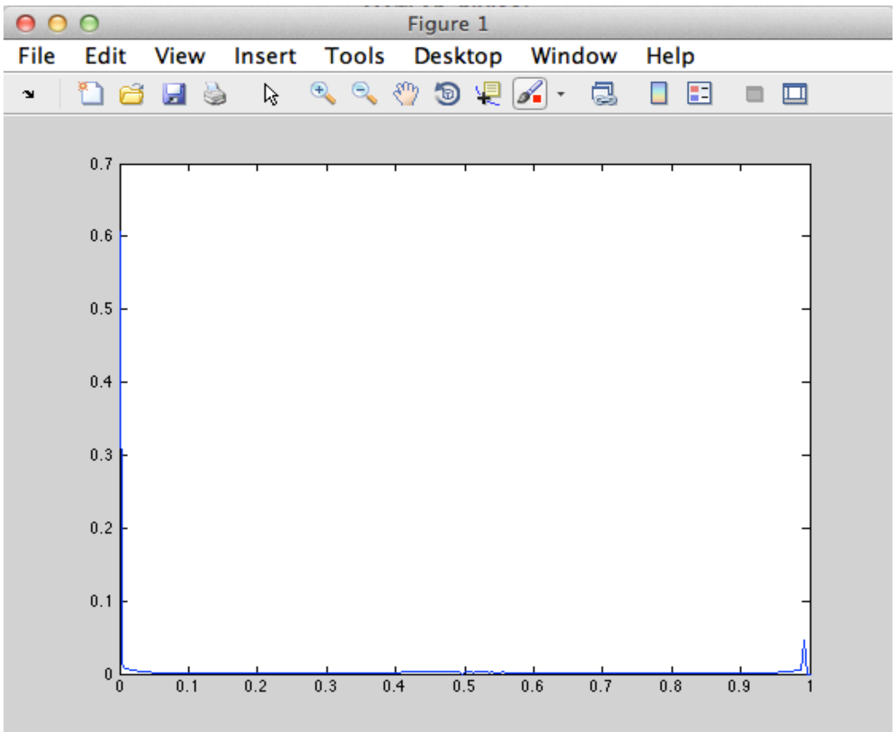
The simple thresholding function works pretty well for the checkerboard images. When the 2nd technique is applied (I took sample of 50x50 in both segments) where we create a Gaussian out of the PDFs and try the same thing, we get the following PDF:



The threshold for this one too is close to 110, which is why it yields almost the same results.

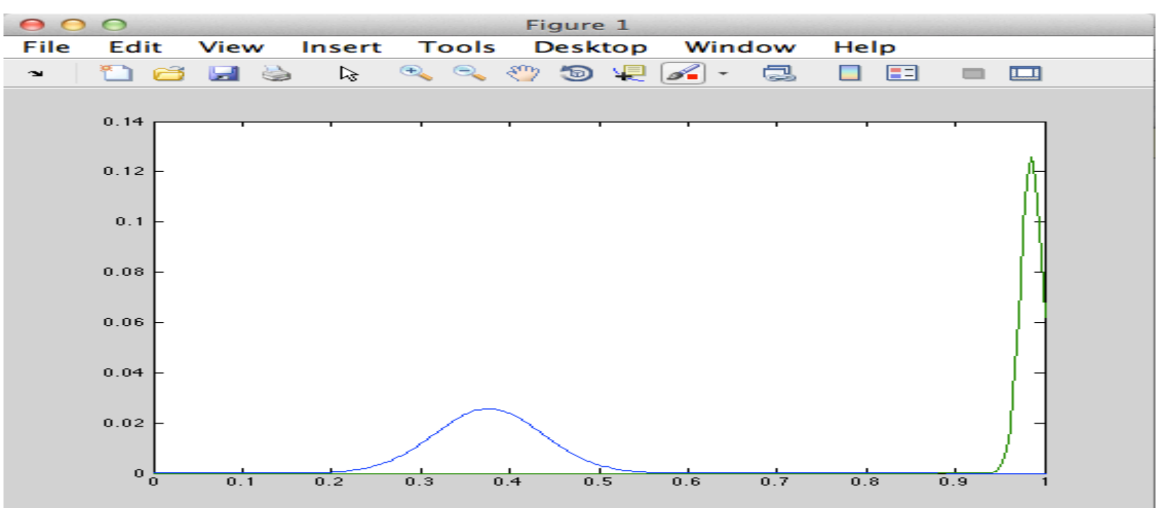
I don't think the original histograms can be recreated from the individual PDFs, since the histogram loses the spatial information and only concerns itself with the frequency of pixels.

In case of the CTscan, the histogram looks like :

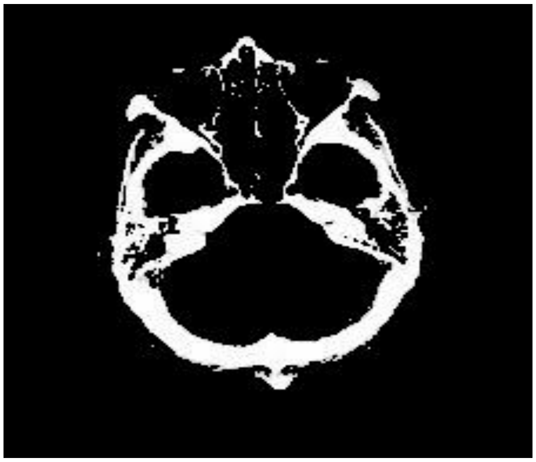


which clearly proves that segmentation based on simple thresholding will not work.

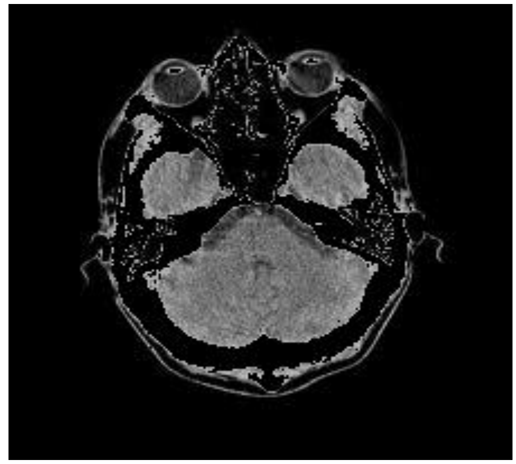
Using the second technique, the Gaussian of the bone and soft tissue is created:



Looking at the distribution above, I made a conservative guess of 0.75 (192) and segmented the scan. The results:



Bone



Soft Tissue

The threshold estimates yields good results proving that the simple thresholding cannot help segment all kinds of images.